A Mixed-Mode Design for a Self-programming Chip For real-time estimation, prediction, and control

Khurram Waheed and Fathi M. Salam

Department of Electrical and Computer Engineering
Michigan State University
East Lansing, MI 48824-1226

Abstract-The paper overviews the development of a self-learning computing chip in 0.18 micron copper technology. This chip supercedes, in its capabilities, present micro-computing paradigms (micro-processors, micro-controllers, and DSPs) in the application domains of process identification, modeling, prediction, and real-time control. In particular, specific domains of targeted potential applications include:

- (i) Nano-level on-line bio-probing and actuation
- (ii) image analysis and feature extraction,
- (iii) channel equalization for high speed mobile communications,
- (iv) inertial navigation sensor fusion, and network management for routers.

I. INTRODUCTION

The core of the chip is a neurally inspired scalable (re-configurable) array network for compatibility with VLSI. The chip is endowed with tested *auto-learning* capability, realized in hardware, to achieve global task auto-learning execution times in the micro to milli seconds. The core consists of basic building blocks of 4-quadrant multipliers, transconductance amplifiers, and active load resistances, for analog (forward-) network processing and learning modules. Super-imposed on the processing network is a digital memory and control modules composed of D-Flip-flops, ADC, Multiplying D/A Converter (MDAC), and comparators for parameter (weight) storage and analog/digital conversions. The architectural forward network (and learning modules) process in analog continuous-time mode, while the (converged, steady state) weights/parameters can be stored on chip in digital form. The overall architectural design also adopts engineering methods from adaptive networks and optimization principles[1].

The 6 level Cu interconnect, single poly, 0.18 micron process enables dense connectivity and dense die area of this highly interconnected network resulting in a compact powerful engine. Moreover, the special low resistance and low capacitance electrical properties of Cu permit the design to achieve the high connectivity while still managing precise distributions of resistive and capacitive loads. These properties enable one to predict performance and limit signal time-delays along the interconnect. The small feature size and the electrical interconnect properties for copper are enablers to the realization of such a powerful chip with dense interconnectivity.

The resulting chip design would require no traditional programming or coding. In addition to novel architectural design, the hardware also performs the heavy computational burden by selectively realizing programmability as on-chip auto-learning modules. The resulting chip super-machine operates on 1.5V power source and would consume less than 1 m Watt..

We mention two prominent potential applications as examples. One example is a smart probe in the medical and biological fields for biological cell measurement and stimulation where no reliable process model exists and where decisions have to be made on-line. Applications in this domain include, drug injections, condition monitoring and micro-surgery. Another example domain is to determine or model the quality of combustion in vehicle engines to detect misfiring and its consequence on exhaust gases and the environment. Both of these application domains generate huge amount of signals or data, and would require massive processing for standard computing paradigms. Similar challenging problems do exist in pattern matching, feature extraction, data mining, to name a few.

II. DESIGN OVERVIEW

Some of the design guidelines we pursued in the design project include:

- 1) The forward network's processing and the learning module is analog, while the weight storage, control signals are digital has given rise to a mixed mode circuit implementation. The multilayer network uses 16 inputs and 16 outputs (see Fig.1).

Fig.1: Block Architecture

- 2) The I/O specification is however flexible and can easily be reduced/expanded in our scalable design to inputs compatible with the available packages. The expansion, however, can be achieved by using several of the chips in cascade and parallel combinations[4].
- 3) The chip operates in four different modes: (i) *learn*, (ii) (on-chip) *store*, (iii) *program read/write*, and (iv) *process* (see Fig. 2)
 - (i) *Learn*: The chip activates the learning process based on the inputs and (desired) output targets supplied by the application or the user.
 - (ii) *Store*: Once the user is satisfied with the performance of the network in the learning mode, the *store* mode saves the computed weights in on-chip static digital memory.
 - (iii) *Program*: This mode was added to give the chip the capability of weight read-out or read-in. The read-in

signifies programming the synapses/weights for applications where the chip has already been trained.

- (iv) *Process*: The chip is thus ready to be used in the *process* mode where the outputs are generated (i.e., computed) by the forward network.

5) There is no speed/clock specification for the processing and learning components of the chip as the speed is set by the application and the architecture's time-constants. In the *learn* mode, the speed is determined by the time the network takes to adapt itself to the input-target patterns. In digital implementations of neural networks, this stage takes the longest time, increasing with the complexity and number of training patterns. For the present chip, and based on our experience with IC chip implementations [2,3], the time it takes to converge to a solution can be around 100-1000 micro seconds-- in conservative 2 micron technology. Storing the (array of) weights takes longer than training, but similar to the *learn* mode, it is only executed once, row by row, in any training session. The *process* mode consumes a delay determined by the largest analog path and the time constant parasitics as all the computation/processing is executed instantaneously and in parallel.

6) The local static memory solution was adopted to counter the weight decay in generic analog implementations of analog neural implementations. On-chip digital memory was preferred as opposed to off-chip memory as it ensures self-contained operation and speed. After the learning phase, all the steady-state analog weights are converted to digital for block-wise storage using the on-chip ADCs to ensure efficient resource utilization and shorter store mode execution.

Fig. 2 : Hierarchical structure of the chip

7) The chip is mixed mode, mixed signal. We call it mixed mode in the sense that the learning phase is pure analog, while the storage mode is analog-digital. This hierarchical view is also maintained in the layout of the chip and in routing interconnects as depicted in Figure 3 below.

The paper will touch upon other aspects such as Architecture Design, High Level design, Topology, Learning algorithm, and Interconnectivity.

III. ARCHITECTURAL DESIGN

A. Synaptic Cell

The processing stage is composed by 16 neurons built using (analog) vector product multipliers and a sigmoid function. The multipliers use as operands an input vector and a weight vector. The input is common to all processing units, and the weights belong to each neuron. The scalar product is then applied to the non-linear function, resulting in the output of a unit neuron.

On-chip memory is designed as local digital memory. It is therefore necessary to add a stage where the present analog value of the weight is converted into a digital value using an ADC, and then converted back using a DAC. The memory is built by using 5 data flip-flops. The update law, however, uses a capacitor (see Fig. 3) and 1-dimension (1-D) multipliers. These multipliers are also used in each neuron, to form the 17-dimension (17-D) multipliers.

To optimize the number of ADCs required for the conversion of the weight and still achieve good performance, a column of ADCs was designed away from the neural network. This design uses multiplexers, decoders, control logic for the *store* mode and the need of a clocked input to drive this logic. This clock would also drive the ADC, as it is designed using the successive approximations method. Observe that having a clock in this section does not imply that the neural network stops being asynchronous.

Fig. 3 : Synaptic Cell Structure and Interconnects

The complete synaptic architecture is pictured in Fig. 3. Now a column represents each neuron layer and each individual element of the array contains a synapse multiplication and a part of the update learning law. The nodes where the grid elements converge compute the sum of the synapses, and proceed to apply a sigmoidal function for the output of the neuron.

Moreover, by storing weights locally in digital format, but still using common ADCs to perform the conversions, a more compact synapse cell is obtained, and the smallest buses are used throughout the chip [6,7].

B. Control Cell

The control cell comprises of the entities contained on the left column of Fig. 3. This houses the successive approximations ADC and the Multiplexer. It is based upon

the MDAC being used in the individual cells. The conversion is achieved by approaching the digital representation in steps, which suggests the use of a clocked logic. In the designed chip the clocks to the D type flip/flops are provided sequentially from the external pins. A PLD / FPGA in a circuits can perform this task very easily. These flip-flops apply a constant digital input to the MDAC for a short time and a feedback signal is computed and used to set the state for the next approximation. Fig. 4 shows the basic synapse cell and the design of the ADC. The multiplexer and decoder are used in tandem to apply the column signals to this ADC one at a time.

The chip has two separate resets, one for the ADC and another for the local weight flip-flops. This, in conjunction with the C12 signals shown below, allow the chip to be programmed (externally downloaded) with pre-determined weights as well.

Fig. 4: Control Cell Structure and Interconnect

IV. SIMULATIONS

A high level matlab simulation was used to verify the overall architectural and learning design. More detailed HSPICE simulations were used to simulate and verify the performance of components as well as sub-circuits of the chip. The HSPICE net lists were also used to verify the layout of the complete chip. All the circuits used in the chip were custom designed and rigorously simulated. The simulation details are beyond the scope of this paper; however, a list of major designed circuits is as follows: 4-Quadrant Gilbert Multipliers; Wide-Transconductance Amplifiers; Multiplying D/A Converter (MDAC); Comparator; Decoders; Multiplexers /De-multiplexer; D-Flip-flops; Logic Gates; Buffers; Transmission Gates; Active Load Resistances; ADC; Vector Multiplier.

V. LAYOUT AND INTERCONNECTS

A. Layout

Fig. 5 below shows the array structure of the implemented chip which shows a highly dense building block structure. The zoomed view (see Fig. 6) gives a better view of the relative placement of the various subcells.

The building block below comprises of identical cells of 17 X 16 synaptic cells (16 inputs and 1 bias) augmented by a column of control cells. The four cells at the bottom are the decoders and de-multiplexers required for chip level programming of synaptic weights for both the blocks in parallel and are used for both row and column selections.

Several of these building blocks can be joined together in series and parallel to obtain a scalable neural structure. In our chip while attaching to the next building block, the intermediate outputs are also routed to the external pins which allows the application of recurrent neural learning structures to the chip.

Figure 5: Layout for the Building Block

A zoomed view of the control and synaptic cells is presented below (Fig. 6). The various building cells such as MDAC, D-flipflops, Comparator, Or gate, Multiplexer and Transmission gates are (included) in the control block while MDAC, Gilbert Multiplier, Local Memory, Temporary Analog Memory and buffers are in the synaptic block.

Fig. 6: Zoomed view of control and synaptic cells

B. Copper Technology & Interconnect

With these new copper chips, we have about 40% less resistance than aluminum, which can translate into a 15% expected speedup of our chips. The power consumption is reduced drastically by 30%, which is of great value to mobile and embedded electronics. The process to manufacture a copper IC actually is expected to cost 20-30% less in volume than that of Al, given that fewer steps are required in its production. With Aluminum quickly reaching its physical limits in semiconductor design we look to copper to take us to the next level of computing platforms.

From the perspective of our project, the UMC 6 level Cu interconnect process enables the achievement of the dense connectivity in our design project. In consequence, it produces dense die area, resulting in a compact powerful neural architectural chip design. Moreover, the special electric Cu properties (low resistance and low capacitance of the interconnect) permit more precise allocation/distributions of resistive and capacitive loads in the chip design of highly connected neural architecture. The small feature size and the Cu interconnect are enablers to the realization of our proposed highly interconnected neural machine. Fig. 7 depicts an example of dense interconnect design in the project.

Fig. 7: Metallic Interconnect Structure of Building Block

A hierarchical metallic routing structure was adopted in the chip layout, which can be better viewed in the zoomed view of control and synaptic cells in Fig. 8. The distribution of metals for the layout is summarized in the table below[6,7].

VI. SUMMARY

The architectural design for the neural network core can be realizable in the Copper technology process for the following reasons:

- (i) small feature size of 0.18 μm , allowing dense, compact and low power design of such a system.
- (ii) copper metal interconnect which made it feasible to use long connections without adversely creating sizable imbalance in resistive and capacitive loads for the interconnected units.
- (iii) the availability of 6 layers of interconnect which allows the designation of layer 1 and 2 for local cell interconnections, layer 3 and 4 for “global” connections,

and layer 5 and 6 for global ground and power source, enabling the dense interconnection for a neural network of this size on VLSI.

Fig. 8 : Zoomed Metal Interconnect view of cells

The proposed chip design impacts several domains of critical applications that include nano-scale biotechnology, automotive sensing, control and actuation, wireless communications, internet routing, image feature extraction and pattern matching etc. These are challenging application domains that presently can not be effectively met by prevalent computing architectures, due to the NP-complete nature of these problems, in the domain of conventional digital computing.

Further details of this design will be provided in future papers.

VII. SELECTED REFERENCES

- [1] Gert Cauwenberghs and M. Bayoumi, (editors) Learning on Silicon, adaptive VLSI neural systems, Kluwer Academic Publishers, July 1999.
- [2] Hwa-Joon Oh and Fathi M. Salam, “Analog CMOS Implementation of Neural Network for Adaptive Signal Processing,” Proc. of The 1994 IEEE International Symposium on Circuit and Systems, London, England, May 30 - June 2, 1994, pp.503-506.
- [3] F. M. Salam, H-J. Oh, “Design of a Temporal Learning Chip for Signal Generation and Classification,” Analog Integrated Circuits and Signal Processing, an international journal, Kluwer Academic Publishers, Vol. 18, No. 2/3, February 1999, pp. 229-242.
- [4] M. Ahmadi and F. Salam, Special Issue on Digital and Analog Arrays, International Journal on Circuits, Systems, and Computers, October/December 1998 (Issue published in December 1999).
- [5] The SRC Website: <http://www.src.org/>
- [6] MSU Team, Copper IC Design Challenge, Phase I Report, August 1999.
- [7] MSU Team, Copper IC Design Challenge, Phase II Report, January 2000.